Evolutionary optimization of the capital and asset structure of the enterprise based on simulation of changes in its market value

Michał Badura[1] Michał Białożyt[2]

[1]AGH, Departament of the Computer Science, Faculty: Data Science
mbadura@student.agh.edu.pl

[2]AGH, Departament of the Computer Science, Faculty: Computer Science
mbialozyt@student.agh.edu.pl

2024

Abstract

Abstract of the paper
Keywords: evolutionary algorithms, neural networks, asset structure, simulation, capital

1 Introduction and review of related research

In our quest to optimize the capital and property structure of enterprises, we leverage machine learning models trained on historical financial data within an agent-based bio-inspired algorithm. This approach enables us to assess individuals and refine decision-making processes in allocating capital and managing assets.

Our research delves into the intricate dynamics of capital and property structures, analyzing the percentage shares of individual components to gain a deeper understanding of resource distribution. Additionally, we examine the internal composition of current assets to gauge liquidity, solvency, and operational efficiency.

By integrating machine learning and bio-inspired principles, our aim is to uncover innovative strategies for enhancing financial management practices, ultimately fostering sustainable growth and resilience in today's economic environment.

In recent years, the integration of artificial intelligence (AI) techniques in corporate finance has gained considerable attention, particularly in the realm of optimizing capital structure decisions. A thorough literature review reveals several significant contributions in this domain. Lee [3] introduced a deep learning framework specifically tailored for determining the optimal capital structure, demonstrating the efficacy of leveraging neural networks in financial decision-making processes. Li [4] conducted a comprehensive survey on the application of machine learning in corporate finance, highlighting its potential in optimizing capital structures among other financial tasks. Imtiaz et al. [2] provided insights into predicting financial distress using machine learning, indirectly addressing the importance of maintaining an optimal capital structure to mitigate such risks. Additionally, Bai et al. [1] offered empirical evidence on the influence of AI on capital structure decisions, emphasizing its growing significance in shaping *corporate financial strategies, particularly in emerging markets like China. Collectively, these studies underscore the transformative role of AI in revolutionizing traditional approaches to capital structure optimization, facilitating more data-driven, efficient, and informed decision-making processes for firms globally.*

2 Problem formulation and proposed solution

2.1 Problem formulation

The optimization of capital and property structures within enterprises is a multifaceted task, requiring the identification of outliers in financial data, prediction of market value changes resulting from structural adjustments, and the search for an optimal configuration using bio-inspired optimization algorithms.

2.2 Proposed solution

1. **Detection of Outliers:**
   - **Task:** Select and train machine learning algorithms to detect outliers in financial data.
   - **Solution:** Implement an anomaly detection algorithm trained on historical financial data to identify outliers in capital and property structures.
2. **Prediction of Market Value Changes:**
   - **Task:** Develop a machine learning model to predict changes in enterprise market value due to alterations in capital and property structures.
   - **Solution:** Train a predictive model using historical data to forecast the impact of structural changes on market valuation.
3. **Search for Optimal Structures:**
   - **Task:** Implement agent-based bio-inspired optimization algorithms to search for the optimal capital and property structure.
   - **Solution:** Develop agent-based versions of bio-inspired optimization algorithms and integrate them with the trained machine learning models for outlier detection and market value prediction. Apply these algorithms iteratively to explore and refine potential configurations, seeking the optimal balance between capital and property elements.

By addressing these tasks and implementing the proposed solutions, we aim to streamline the optimization process, enhance decision-making capabilities, and ultimately achieve a more robust and efficient capital and property structure within enterprises.

3 Data Processing and Model Training

The data source section is foundational for establishing the framework necessary to implement the machine learning models described. Our research focuses on the utilization of historical financial data extracted from company balance sheets and market value indicators, aimed at enabling precise modeling and analysis within an agent-based bio-inspired algorithm. This algorithm is structured to explore and identify the optimal capital and property structure of enterprises.

3.1 Financial Data Structure

The core data comes from company balance sheets which are systematically organized and readily available in an Excel format. This data is pivotal for understanding the composition of assets and liabilities of a company over time. For our analysis, we will specifically leverage the annual data (YS tab) and quarterly data (QS tab) from the "Balance sheet" section as depicted in the provided assets-and-liabilities-example.xlsx file. This structured format allows us to directly apply our machine learning models to assess and forecast the financial health and potential of enterprises.

3.2 Market Value Data

Additionally, the market value of enterprises is another crucial component of our dataset. This value is particularly important when assessing the financial impact of changes in capital and property structures on the overall market perception and valuation of a company. For our purposes, data will be gathered from the Warsaw Stock Exchange (WSE) yearbooks, which list enterprise market values annually. For instance, market values for companies like Budimex and Budopol in specific years provide a benchmark for analyzing trends and shifts in enterprise valuation over time. Moreover, online resources such as the WSE website and other financial platforms like https://stooq.pl/ will be utilized to supplement and update our market value data as needed.

4 Data processing and model training

Whole concept of data processing and model training for predicting market values of companies is included in Jupyter notebook script. Below are the individual stages of the script:

4.1 Importing Necessary Libraries

The script begins by importing the required libraries such as `pandas` for data manipulation, `scikit-learn` for machine learning models, and functions from a custom script (`import_data`).

4.2 Importing Market Values Data

Market values data is imported from a CSV file named `market_values.csv`. The data is then set to have the index as the first column.

4.3 Importing Balance Sheet Data

Balance sheet data for companies is imported using a custom function import_csv_dict_of_dataframes. This function seems to read data from a CSV file containing balance sheet information for multiple companies.

4.4 Data Extraction

Using nested loops, the code iterates through each company and each year within the company's data. A new dictionary, `new_row`, is created for each year of data. It checks each indicator in the company's financial data against the predefined list of `columns`. If the indicator exists, its value is directly assigned; otherwise, it's set to zero. This step ensures all rows have uniform columns, filled with actual values or zeros.

4.5 Market Value Integration

The market value for each company and year is retrieved from a separate dataset, `market_values`. If the company and year are present in the `market_values` index, the corresponding market value is added to `new_row`.

4.6 DataFrame Formation

All `new_row` dictionaries are aggregated into the `rows` list, which is then converted into a pandas DataFrame. This structure allows for more complex manipulations and cleaning processes.

4.7 Handling Missing Values

The DataFrame is then processed to replace all NaN values with zeros using the `fillna(0)` function, ensuring no missing data points that could disrupt further analysis.

4.8 Data Type Standardization

The `apply(pd.to_numeric, errors='coerce')` function is used to convert all entries into numeric types, coercing errors by converting problematic non-numeric entries to NaN, which are then filled with zeros.

4.9 Filtering Invalid Entries

Rows where the `ASSETS` or `market_value` are zero or negative are removed, as these do not represent valid data for analysis.

4.10 Normalization

Finally, the asset-related columns (excluding the first) are normalized against the total assets (`ASSETS`) to express each as a percentage of total assets. This normalization is crucial for comparing companies of different sizes on a like-for-like basis.

4.11 Splitting Data into Features and Target

The dataset is split into features (`X`) and the target variable (`y`), where `X` contains all columns except `'market_value'` and `y` contains only the `'market_value'` column.

4.12 Model Training and Evaluation

The dataset is divided into training and testing subsets using the train_test_split function. Various machine learning models such as Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, Support Vector Machine, and Neural Network are initialized and trained on the training data. The performance of each model is then evaluated based on the Mean Squared Error (MSE) metric, with the results printed out for analysis.

- **Linear Regression:**
  - MSE: 47,787.793
  - This model shows a relatively high MSE, suggesting less accuracy in fitting the data compared to other models.
- **Random Forest:**
  - MSE: 35,647.288
  - The improvement in MSE over Linear Regression indicates a better fit and more accurate predictions.
- **Gradient Boosting:**
  - MSE: 46,911.567
  - This model performs better than Linear Regression but still has a higher MSE compared to other models, showing moderate effectiveness.
- **Support Vector Machine (SVM):**
  - MSE: 22,330.285
  - SVM exhibits the lowest MSE among the models evaluated, suggesting a potentially good fit and high effectiveness for this dataset.
- **Neural Network:**
  - MSE: 24,530.492
  - The Neural Network also shows a low MSE, indicating potentially good performance and the ability to capture complex patterns in the data.

Conclusion: The analysis of MSE values shows that the Support Vector Machine and Neural Network models have the best performance, making them more suitable for this specific dataset. These results underscore the importance of model selection based on data characteristics and desired accuracy.

Overall, this script demonstrates a standard pipeline for data preprocessing and model training for predicting market values of companies using various machine learning algorithms.

5 Evolutionary algorithm

An evolutionary algorithm (EA) simulates the processes of biological evolution to find optimal solutions to optimization problems. Below is a detailed description of the EA implementation used to optimize the capital structure of a company.

5.1 Population Initialization

The first step is to initialize the population. The population consists of individuals, each representing a potential solution to the problem. In our case, each individual is a vector of values representing the proportions of capital allocated to different company assets. The initial population is generated randomly, and the proportions are scaled to sum up to 100

5.2 Population Evaluation

Each individual in the population is evaluated based on the market value difference, which can be estimated using one of the trained models. The result of this evaluation is a fitness value.

5.3 Parent Selection

Selecting parents for crossover is a crucial element of the algorithm. Parents are chosen from the population with a probability proportional to their fitness, increasing the chance of passing on better traits to the offspring.

5.4 Crossover and Mutation

The crossover process creates new individuals (offspring) by combining traits of selected parents. Mutation introduces random changes, increasing the genetic diversity of the population. Crossover involves randomly combining parts of the parents' genomes, while mutation introduces small, random changes to the offspring's genome. In our algorithm crossover takes several properties from one parent and the rest from the second one. In the end all values are scaled appropriately to make all properties sum up to 100%. 2 different versions of mutation have been introduced. One mutation randomly changes the value of a property (and then performs scaling to maintain the 100% sum), second mutation swaps 2 random values among all properties.

5.5 Selection of the Next Generation

The next generation is formed from the best-adapted individuals of the current population and their offspring. This process allows for the gradual improvement of the population's quality over successive generations.

5.6 Main Loop of the Algorithm

The main loop of the algorithm includes population initialization, population evaluation, parent selection, crossover, mutation, and selection of the next generation. This process is repeated for a specified number of generations until an optimal solution is obtained. The best fitness is monitored and visualized across generations, allowing for an assessment of the algorithm's progress.

5.7 Parameters

Values such as population size, number of generations and mutation rate should be appropriately set. For the experiments the following have been selected

- population size: 500
- number of generations: 400
- mutation rate: 5% (both version equally probable)

6 Results

In this section we show and compare simualation results conducted for each model that was previously trained. Each section contains a plot showing how was best population fitess changed during the algorithm run as well as the analysis of best performing population.

6.1 Linear Regression

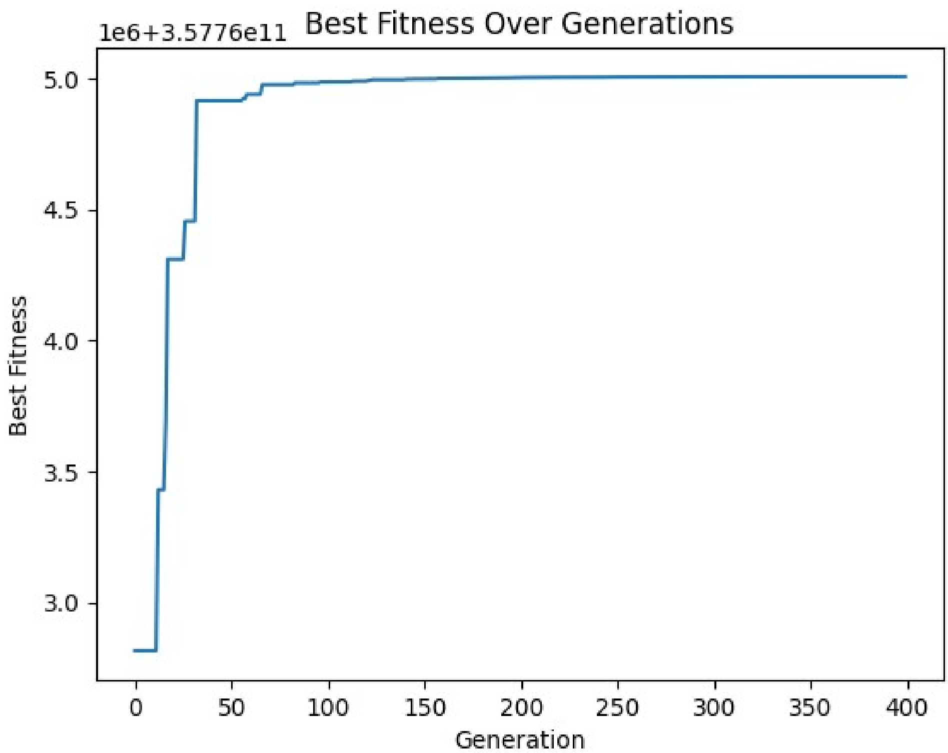

Figure 1: Best fitness over generation number for linear regression

Capital property components with highest value:

- Liabilities related to assets held for sale and discontinued operations: 99.98%
- Provisions: 0.001%
- Inventories: 0.001%
- Current liabilities: 0.001%
- Non-current loans and receivables: 0.001%

Linear regression is clearly not the best model for evaluating the capital property structure of a company. It uses a linear function and simply maximizes the variable that theoretically should most boost the company's growth (the variable with the highest slope). This approach results in an absurdly high fitness value. Unfortunately, such a scenario is unrealistic and could never be applied in any company.

6.2 Random forest regressor

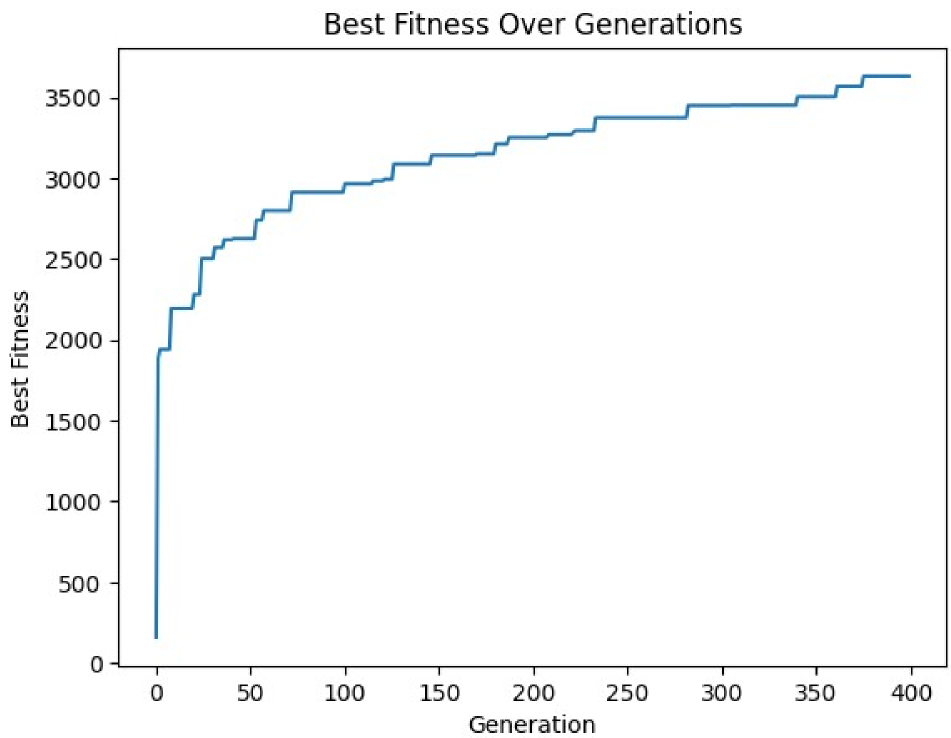

Figure 2: Best fitness over generation number for random forest regressor

Capital property components with highest value:

- Property, plant and equipment: 40.77%
- Other liabilities: 5.56%
- Loans and other receivables: 5.25%
- Retained earnings / accumulated losses: 4.67%
- Current tax liabilities: 3.45%

The random forest regressor produces much more reliable results. In numerous training data examples, the "property, plant, and equipment" asset ranges between 35% and 45%. This indicates that the random forest classifier model could potentially be useful for such problems. The curve shown in Figure 2 suggests that the simulation might have continued to improve. Adding another 400 generations could potentially optimize the structure even further.

6.3 Gradient boosting

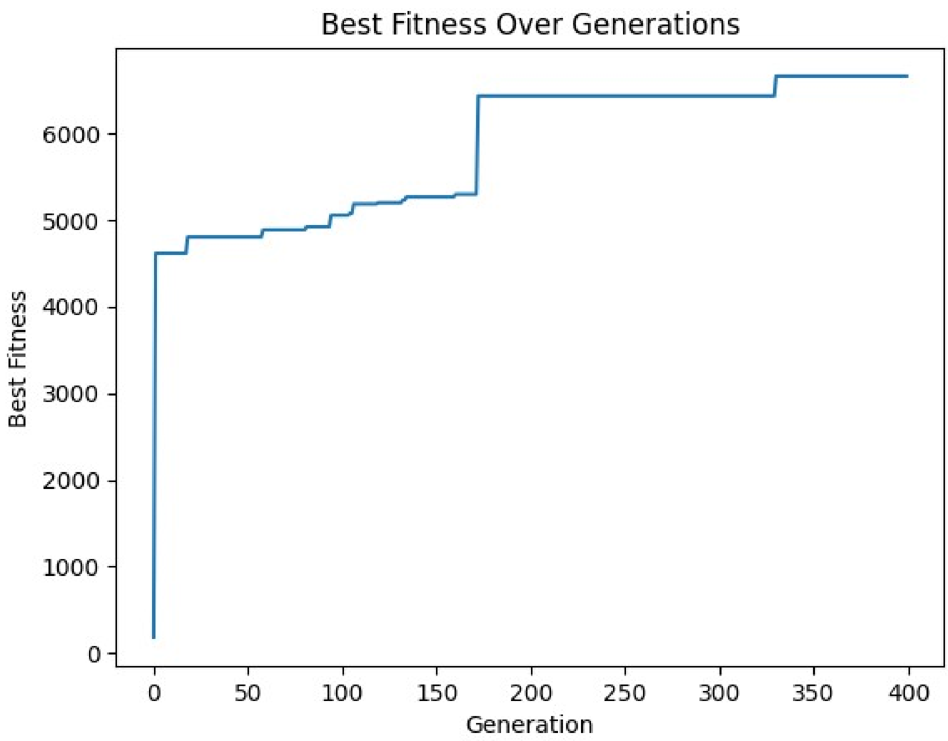

Figure 3: Best fitness over generation number for gradient boosting model

Capital property components with highest value:

- Equity shareholders of the parent: 31.21%
- Current tax liabilities: 22.19%
- Non-current loans and borrowing: 16.53%
- Trade payables: 5.11%
- Goodwill: 3.67%

Gradient boosting regressor simulations yielded significantly different results compared to previous models. The best fitness achieved was approximately twice as high as that of the random forest classifier, although these conclusions should be approached with caution. The linear regression model produced extremely high fitness results but proved entirely impractical for real-world applications. It is important to emphasize the focus on high tax liabilities in these results.

6.4 Support Vector Machines

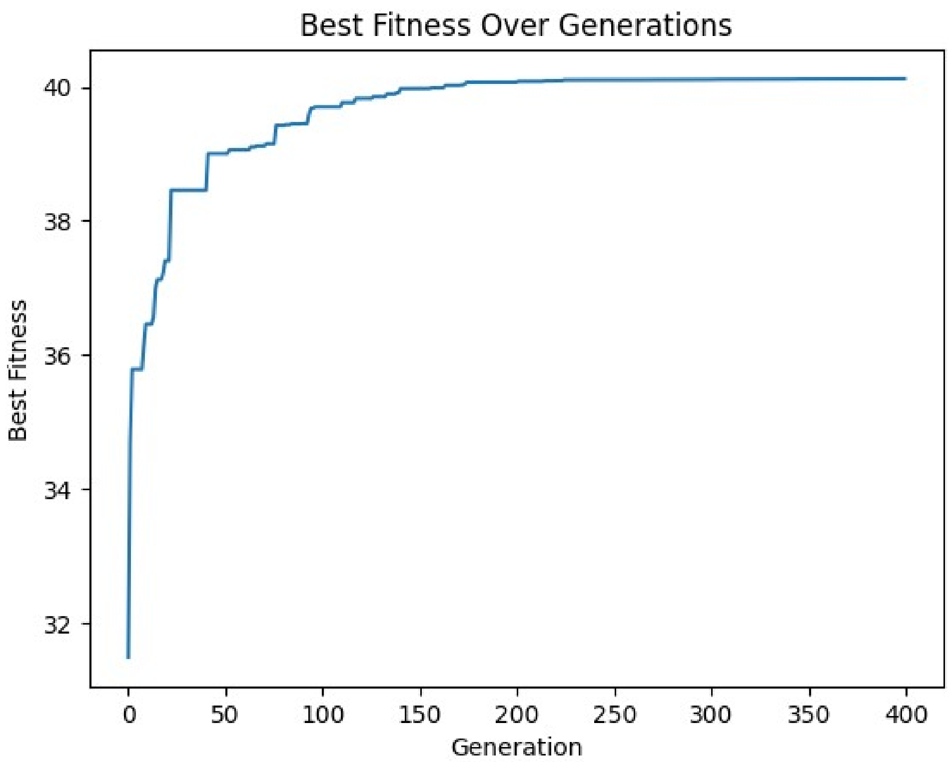

Figure 4: Best fitness over generation number for support vector machines model

Capital property components with highest value:

- Cash and cash equivalents: 99.91%
- Non-current liabilities: 0.01%
- Non-current liabilities from finance leases: 0.01%
- Non-controlling interests: 0.01%
- Called up capital: 0.01%

The support vector machine model performed poorly in evaluating the capital property structure. Allocating all funds to cash and cash equivalents is likely not the best solution. The best fitness value generated was suboptimal. It is recommended to thoroughly analyze the algorithm to determine if it became stuck in a local minimum.

6.5 Deep neural network

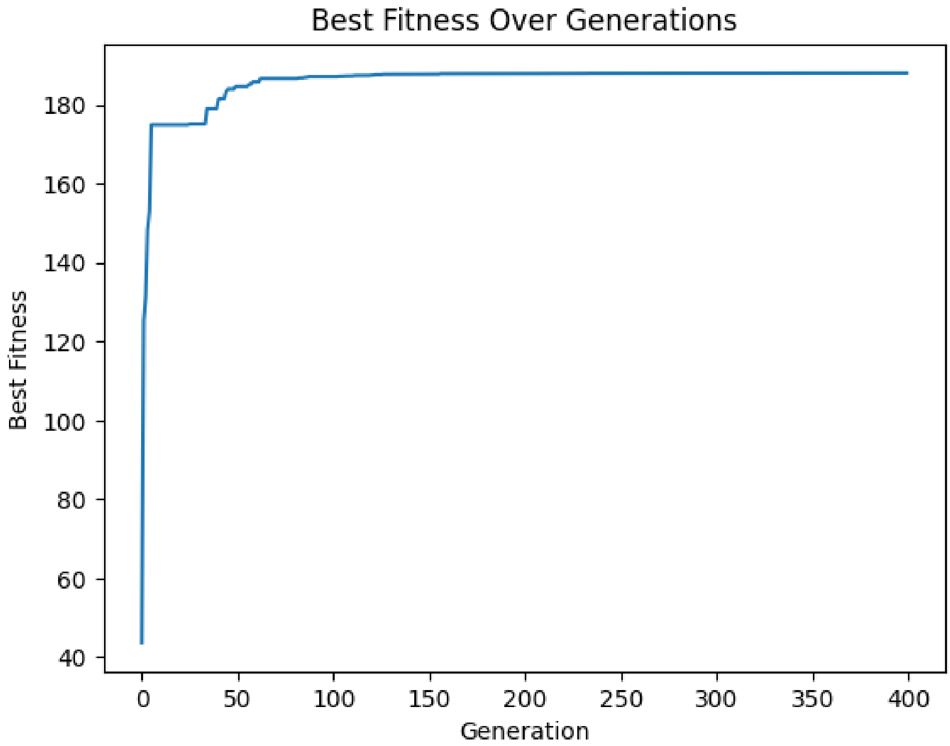

Figure 5: Best fitness over generation number for deep neural network

Capital property components with highest value:

- Cash and cash equivalents: 81.29%
- Non-current liabilities: 18.66%
- Non-current loans and borrowings: 0.01%
- Deferred income tax: 0.01%
- Supplementary capital: 0.01%

The neural network used was a 3-layer MLP regressor with 100 neurons in the hidden layer. The optimal strategy produced by the deep neural network simulations consisted entirely of cash and non-current liabilities. The best fitness value achieved by this simulation was much lower than those from the random forest regressor and gradient boosting regressor. The simulation reached its optimal point early, around the 100th generation. There is a possibility that it became stuck in a local minimum.

7 Conclusions

Evolutionary algorithms offer a powerful tool for optimizing complex problems, such as a company's capital structure. Through the mechanisms of selection, crossover, and mutation, it is possible to find optimal solutions that maximize the firm's market value. The implementation described above demonstrates how these techniques can be practically applied, bringing significant benefits in the context of capital management. Each model trained produced completely different results, some of which were immediately recognizable as impractical for real-world scenarios.

The evolutionary algorithm based on machine learning models cannot precisely predict a company's growth, as this is a highly complex process influenced by numerous factors. A company's property structure is just one of hundreds of factors affecting its performance. Sometimes, the financial situation forces management to change the capital property structure in ways they would not normally prefer.

The models that produced the most reliable results were the gradient boosting regressor and the random forest regressor. The evolutionary algorithm based on the random forest regressor resulted in a capital property structure very similar to some of the examples it was trained on. This indicates that these algorithms have high potential to be useful with further research. However, it is difficult to judge if the trained machine learning models are optimal and can effectively predict a company's growth. Additional research, including other factors that can affect the final result, is required.

References

[1] L. Bai. “"Artificial Intelligence and Capital Structure Decisions: Evidence from China"”. In: *Finance Research Letters* (2021).

[2] A. Imtiaz. “"Predicting Corporate Financial Distress Using Machine Learning: A Systematic Literature Review"”. In: *Expert Systems with Applications* (2021).

[3] M. Lee. “"A Deep Learning Framework for Optimal Capital Structure Decisions"”. In: *Journal of Financial Management* (2019).

[4] Y. Li. “"Machine Learning in Corporate Finance: A Survey"”. In: *Pacific-Basin Finance Journal* (2019).